Datové struktury I

9. přednáška: Bloom filtry

Jirka Fink

https://ktiml.mff.cuni.cz/~fink/

Katedra teoretické informatiky a matematické logiky
Matematicko-fyzikální fakulta
Univerzita Karlova v Praze

Zimní semestr 2024/25

Bloom filtry (Bloom, 1970)

Cíl

- Chtěli bychom datovou strukturu, která umí přidávat prvky a zjišťovat, zda byl daný prvek vložen
- Máme dlouhé klíče a všechny se nám nevejdou do paměti ①
- Nevadí nám, když datová struktura občas vrátí špatnou odpověď

Postup (varianta s 1 tabulkou)

- Máme množinu S velikosti n z univerza U
- Použijeme bitové pole M velikosti m ②
- Zvolíme k hešovacích funkcí $h_1, \ldots, h_k : U \to M$ ③
- Na počátku jsou všechny bity nulové
- Při vložení prvku $x \in S$ nastavíme bity $h_1(x), \ldots, h_k(x)$ na jedničku
- Jestliže pro $y \in U$ je některý z $h_1(y), \ldots, h_k(y)$ nulový, pak určitě y nebyl vložen
- Jestliže jsou všechny bity $h_1(y), \ldots, h_k(y)$ jedničkové, tak si nemůžeme být jisti, že prvek nebyl vložen

Varianta s k tabulkami

Máme k bitových polí velikosti $\frac{m}{k}$ a funkce h_i nastavuje bity v poli M_i.

1. Klíče můžou být uživatelem navštívené url adresy nebo všechna analyzovaná řešení genetického algoritmu. nedostatečná kapacita paměti může být způsobena obrovským množství dat nebo omezeným množstvím paměti, například v sušenkách prohlížečů.
2. Pamatujeme si jen m bitů, nikoliv n klíčů.
3. Pro potřeby analýzy budeme předpokládat, že hešovací systém je úplně nezávislý.

Bloom filtry: Jedna hešovací funkce

Cíl

- Jaká je pravděpodobnost, že dostaneme kladnou odpověď, i když prvek nebyl vložen?
- Jak zvolit počet funkcí k, abychom minimalizovali pravděpodobnost špatné odpovědi?

Pozorování

Pro jednu funkci z 1-universálního hešovacího systému do jednoho pole velikosti m při n uložených prvcích máme pravděpodobnost chybné odpovědi testu nejvýše $\frac{n}{m}$. ①

Příklad

- Máme $n = 10^6$ prvků
- Chceme mít pravděpodobnost chybné odpovědi nejvýše $\epsilon = 0.01$
- Potřebujeme tabulku velikosti $m = 100$ Mb.

1. Ptáme se, zda prvek $y \in U$ je v množině S. Pokud je uložený, tak určitě dostaneme pozitivní odpověď. Pokud není, tak pozitivní odpověď dostaneme, pokud existuje $x \in S$ takový, že $h(x) = h(y)$, což nastane s pravděpodobností $P[\exists x \in S :\ h(x) = h(y)] \leq \sum_{x \in S} P[h(x) = h(y)] \leq \frac{n}{m}$.

Bloom filtry: Více tabulek

Pozorování

Pro $k = \lceil \log_2 1/\epsilon \rceil$ funkcí z $(k, 1)$-nezávislého hešovacího systému do k polí velikostí $2n$ při n uložených prvcích máme pravděpodobnost chybné odpovědi testu nejvýše ϵ.

Důkaz

- Pravděpodobnost kolize v jedné dané tabulce je nejvýše $1/2$
- Pravděpodobnost kolizí ve všech tabulkách je nejvýše $\frac{1}{2^k} = \frac{1}{2^{\log_2(1/\epsilon)}} = \epsilon$

Příklad

- Máme $n = 10^6$ prvků
- Chceme mít pravděpodobnost chybné odpovědi nejvýše $\epsilon = 0.01$
- Počet tabulek je $k = 7$
- Potřebujeme tabulky celkové velikosti 14 Mb.

Bloom filtry: Více hešovacích funkcí do jedné tabulky

Pozorování

Pro úplně nezávislý hešovací systém, n prvků a tabulku velikosti m lze volit počet hešovacích funkcí $k = \frac{m}{n} \ln 2$, a poté pravděpodobnost chybné odpovědi je nejvýše 2^{-k}.

Analýza

- Pravděpodobnost, že pozice $h_1(y)$ je nulová je $\left(1 - \frac{1}{m}\right)^{kn}$ ①
- $\left(1 - \frac{1}{m}\right)^{kn} = \left(\left(1 + \frac{-1}{m}\right)^m\right)^{\frac{kn}{m}} \geq e^{-\frac{kn}{m}} =: p$ ②
- Pravděpodobnost, že všechny bity $h_1(y), \dots, h_k(y)$ jsou 1, je nejvýše $(1-p)^k$
- Chceme najít k minimalizující $(1-p)^k = e^{k \ln(1-p)}$ ③
- Z $k = -\frac{m}{n} \ln p$ plyne $k \ln(1-p) = -\frac{m}{n} \ln(p) \ln(1-p)$
- Ze symetrií funkcí $\ln(p) \ln(1-p)$ odhadneme, že maximum nastane uprostřed pro $p = \frac{1}{2}$ ④
- Tedy $k = \frac{m}{n} \ln 2$
- Pravděpodobnost „false positive“ je nejvýše $(1-p)^k = 2^{-\frac{m}{n} \ln 2} = 2^{-k}$

1. Pravděpodobnost, že $h_i(x) \neq h_1(y)$ je $\frac{1}{m}$. Funkcí h_i je k, prvků $x \in S$ je n a náhodné veličiny $h_i(x)$ jsou nezávislé.
2. Z matematické analýzy víme, že $\lim_{m\to\infty}(1+\frac{a}{m})^m = e^a$. Předpokládáme, že $\frac{n}{m}$ je konstanta, jak později uvidíme, že k je též konstanta. Proto je exponent $\frac{kn}{m}$ konstantní. Z matematické analýzy bychom měli vědět, že chyba v aproximaci je $\mathcal{O}(\frac{1}{m})$.
3. Funkce e^a je rostoucí v a, takže stačí minimalizovat expoment.
4. Ve formálním důkazu bychom našli lokalní optima pomocí derivace a ověřili, že máme globální maximum.

Bloom filtry: Více hešovacích funkcí do jedné tabulky

Pozorování

Chceme-li pro úplně nezávislý hešovací systém a n prvků mít pravděpodobnost chyby nejvýše ϵ, pak můžeme volit $k = \lceil \log \frac{1}{\epsilon} \rceil$ hešovacích funkcí do tabulky velikosti $m = \frac{kn}{\ln 2}$.

Analýza

- Pravděpodobnost chyby ϵ jsme odhadli $(1-p)^k$, kde $p = \frac{1}{2}$
- Vyjádřejím k dostáváme $k = \lceil \ln \frac{1}{\epsilon} \rceil$
- Z optimální volby $k = \frac{m}{n} \ln 2$ vyjádříme $m = \frac{kn}{\ln 2}$

Counting Bloom filtry (Fan et al. 2000)

Cíl

Chtěli bychom v Bloom filtru umět mazat prvky.

Postup

- Na každé pozici v tabulce nebudeme mít jeden bit ale malý čítač
- Při operace INSERT se čítače $h_1(x), \dots, h_k(x)$ zvýší o jedna
- Při operaci DELETE se čítače $h_1(x), \dots, h_k(x)$ sníží o jedna
- Jestliže některý z čítačů $h_1(y), \dots, h_k(y)$ je nulový, pak y není přítomen
- Zvolíme $k = \frac{m}{n} \ln 2$
- Jestliže všechny čítače $h_1(y), \dots, h_k(y)$ jsou kladné, pak y není přítomen s pravděpodobností přibližně 2^{-k}
- Pravděpodobnost přetečení některého čítače spočítáme na cvičení

Srovnání hešování

Popis experimentu

Jak reprezentovat graf pro Bellmanův–Fordův algoritmus hledající nejkratší cestu v grafu, jehož hrany mohou být ohodnoceny i záporně? ①

Dict (MetaGraphNext): `Dict{Tuple{Int,Int},Int}`

Prop (MetaGraph): `Dict{Tuple{Int,Int},PropDict}`

Sparse (SimpleWeightedGraphs): Compressed Sparse Column Matrix

Tuple `Vector{Vector{Tuple{Int,Int}}}`
Sousední vrchol a váha pro každou hranu

Výsledky

Vrcholů	Hran	Dict	Prop	Sparse	Tuple
214	45 576	2.2 ms	14 ms	0.480 ms	0.058 ms
1 024	387 396	30 ms	214 ms	13 ms	0.48 ms
3 125	1 274 018	117 ms	907 ms	50 ms	1.7 ms
1 000	151 680	7.7 ms	73 ms	4.2 ms	0.16 ms
1 000	998 980	86 ms	760 ms	17 ms	1.1 ms

1. Různé knihovny pro reprezentaci grafů v jazyce Julia.
   Všechny reprezentace mají uložené pole sousedů pro každý vrchol, ale liší se způsobem uložení vah.
   Datasety: `https://doi.org/10.4121/uuid:653d9ff7-3602-4db1-be71-44dca76a04fa`, `https://doi.org/10.4121/uuid:fea4f033-6bc1-46d8-81f4-d17115304fdb`